Department of Computer Science & Engineering, SDMCET, Dharwad-2

AOOP Assignment Submission Report

[Submitted as part of CTA Assignment No-1

Course:	Advanced Object-Oriented Programming	Course Code:	18UCSE508
Semester:	V	Division:	B

xSubmitted by:

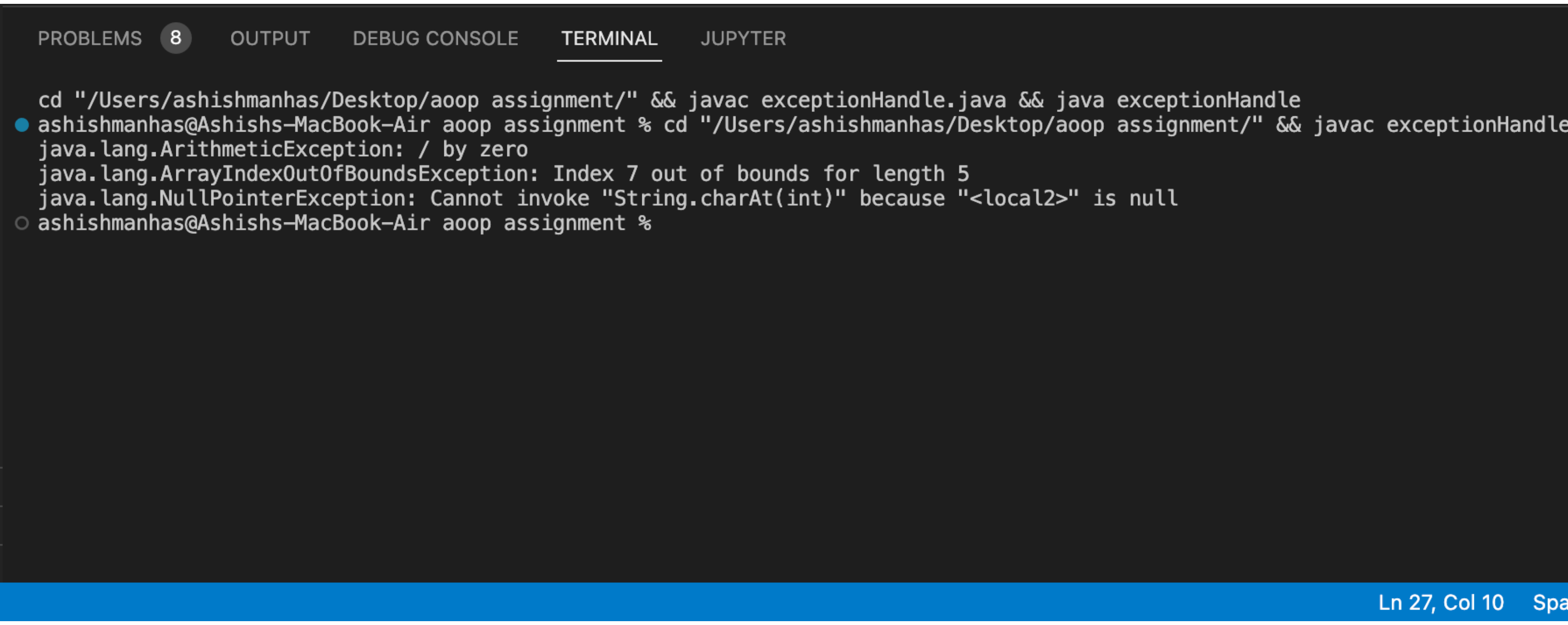
USN: 2SD20CS027 Name: ASHISH MANHAS

1. Problem Definition:

Write a Java program to generate and handle any three built-in exceptions and display appropriate error messages.

```
public class exceptionHandle {
  public static void main (String [] args){
  //arthemetic exception
  try{
  int a=10,b=0;
  int c;
  c=a/b;
  }
  catch(ArithmeticException e){
   System.out.println(e);
  //array index out of bound
  try{
  int a[]=new int[5];
  a[7]=9;
  System.out.println(a[7]);
  }catch(ArrayIndexOutOfBoundsException ae){
   System.out.println(ae);
  }
   //null pointer Exception
  try {
  String a = null; // null value
  System.out.println(a.charAt(0));
   }
  catch (NullPointerException aee) {
  System.out.println(aee);
}
}
}
}
```

Output -:



```
 cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac exceptionHandle.java && java exceptionHandle
● ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac exceptionHandle
 java.lang.ArithmeticException: / by zero
 java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 5
 java.lang.NullPointerException: Cannot invoke "String.charAt(int)" because "<local2>" is null
○ ashishmanhas@Ashishs-MacBook-Air aoop assignment %
```

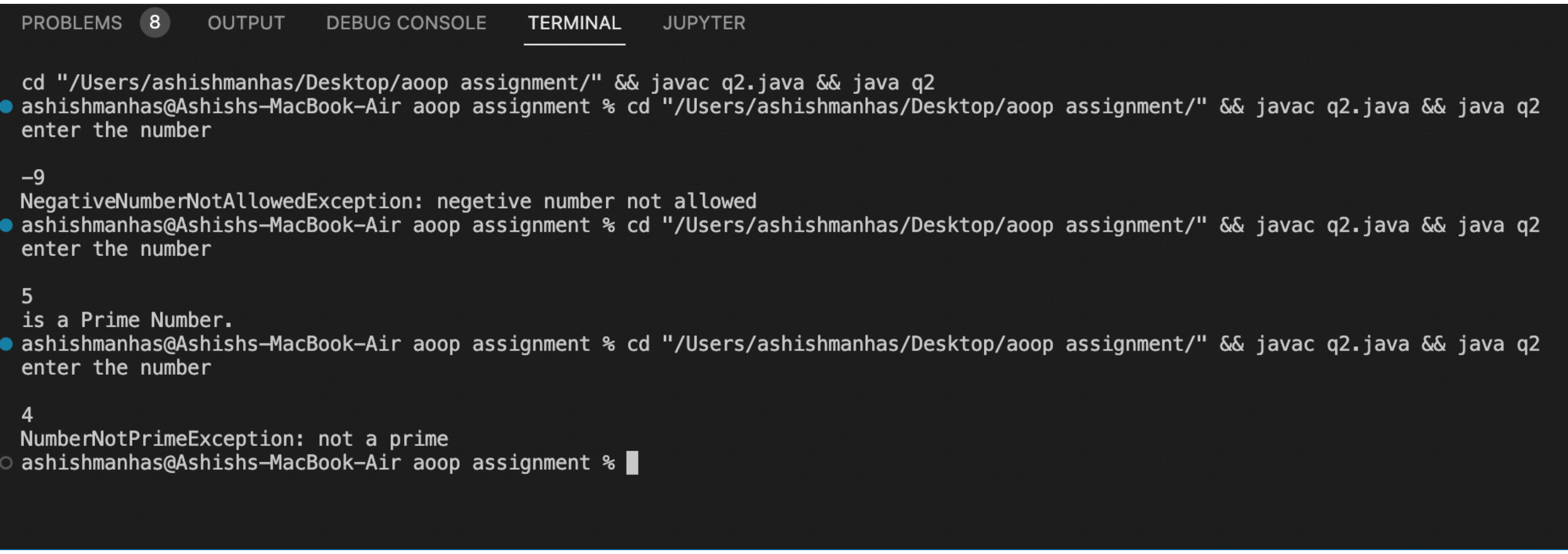
2.Problem Definition:

Write a Java program to read an integer and check whether the number is prime or not. If negative number is entered, throw an exception NegativeNumberNotAllowedException and if entered number is not prime, then throw NumberNotPrimeException.

```
//import java .io.*;
  import java.util.Scanner;
  class NegativeNumberNotAllowedException extends Exception{
  public NegativeNumberNotAllowedException(String st){
  super(st);
  }
 }
 class NumberNotPrimeException  extends Exception{
  public NumberNotPrimeException(String s2){
      super(s2);
  }
 }

 class q2 {
 public static void main(String[]args){
 int i;
 Scanner sc=new Scanner(System.in);
 System.out.println("enter the number\n");
 int n=sc.nextInt();
 if(n<0){
 try {
      throw new NegativeNumberNotAllowedException("negetive

  number not allowed”);

  } catch (NegativeNumberNotAllowedException e) {
      System.out.println(e);
  }
}
```

```
  else{
  int flag =0;
 for(i=2; i<n; i++)
{
     if(n%i == 0)
     {
        flag=1;
        break;
     }
    }
   try{
  if(flag==1) throw new NumberNotPrimeException("not a prime");

     else{
        System.out.println("is a Prime Number.");
     }
    }
    catch(NumberNotPrimeException ex){
        System.out.println(ex);
    }


    }
    }
    }
```

Output -:



```
cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q2.java && java q2
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q2.java && java q2
enter the number

-9
NegativeNumberNotAllowedException: negetive number not allowed
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q2.java && java q2
enter the number

5
is a Prime Number.
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q2.java && java q2
enter the number

4
NumberNotPrimeException: not a prime
ashishmanhas@Ashishs-MacBook-Air aoop assignment % 
```

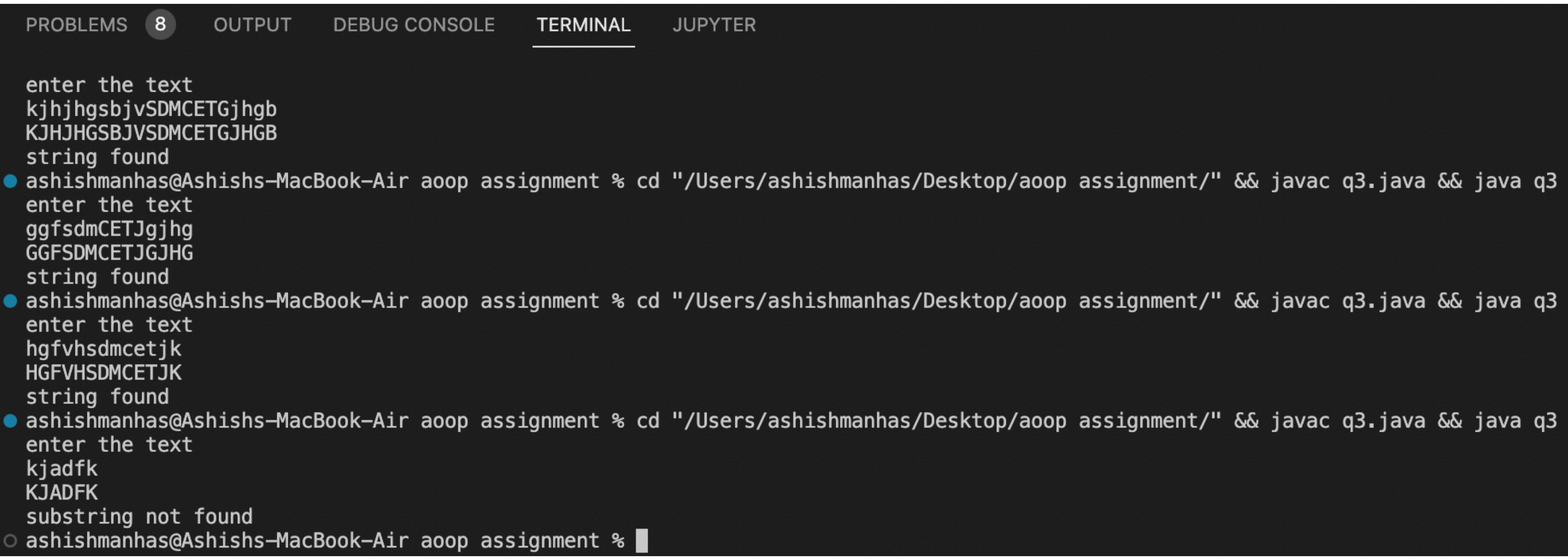
3.Problem Definition:

Write a Java program to perform the following operations:

a) Read a line of text
b) Search for a sub-string SDMCET (case insensitive search)
c) If found, then print success message
d) Otherwise throw an exception SubStringNotFoundException with appropriate message

```
import java.util.Scanner;
class SubStringNotFoundException extends Exception{
 }
public class q3 {
public static void main(String[]args) {
   Scanner sc=new Scanner(System.in);
   System.out.println("enter the text");
   String s1=sc.next();
   String s3=s1.toUpperCase();

   System.out.println(s3);
   String s2 = "SDMCET";
   boolean result = s3.contains(s2);
   if(result==false){
       try{
           throw new SubStringNotFoundException();
       } catch (SubStringNotFoundException e) {
           System.out.println("substring not \
          found");
       }

      }
else{
System.out.println("string found");
}
}
}
```

Output -:



```
enter the text
kjhjhgsbjvSDMCETGjhgb
KJHJHGSBJVSDMCETGJHGB
string found
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q3.java && java q3
enter the text
ggfsdmCETJgjhg
GGFSDMCETJGJHG
string found
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q3.java && java q3
enter the text
hgfvhsdmcetjk
HGFVHSDMCETJK
string found
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac q3.java && java q3
enter the text
kjadfk
KJADFK
substring not found
ashishmanhas@Ashishs-MacBook-Air aoop assignment % 
```

4.Problem Definition:

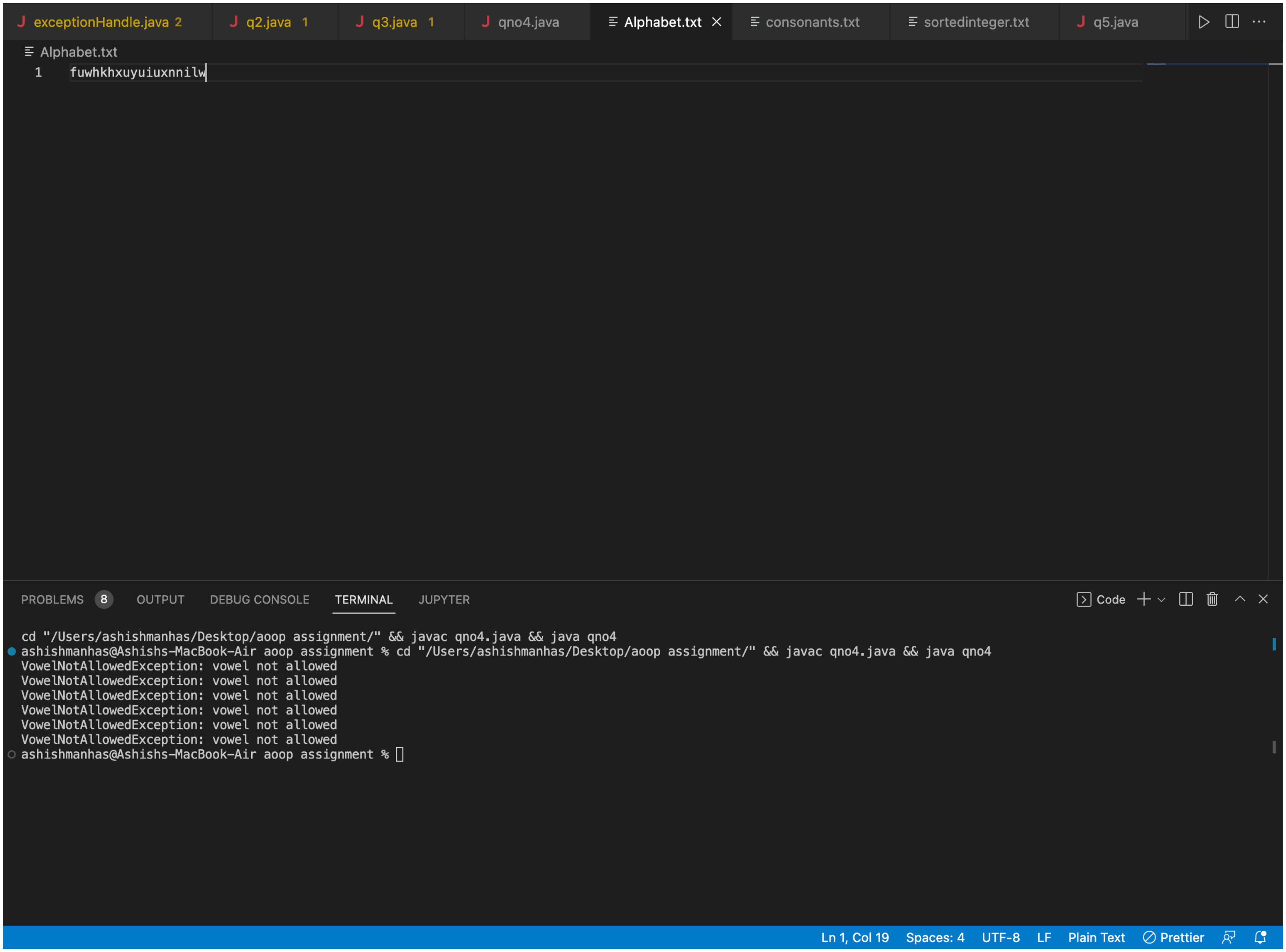
Write a Java program to perform the following operations:

a) Create a file named Alphabets.txt and insert appropriate data into it
b) Read the file and copy all the consonants into another file named Consonants.txt
c) If vowel is encountered, throw an exception VowelNotAllowedException and continue until end of file

```
import java.io.*;
class VowelNotAllowedException extends Exception{
public VowelNotAllowedException(String st){
super(st);
 }
 }
public class qno4 {
public static void main(String[] args)throws
IOException
{
try
{
FileInputStream fstream = new
FileInputStream("Alphabet.txt");
DataInputStream in = new DataInputStream(fstream);
BufferedReader br = new BufferedReader(new
InputStreamReader(in));
FileWriter opstream = new
FileWriter("consonants.txt");
BufferedWriter outt = new BufferedWriter(opstream);
String str=br.readLine();
```

```
    char ch;
    for (int I=0;i<str.length();i++)
    {
    ch = str.charAt(i);
  if(( ch == 'a') ||( ch == 'e') ||( ch == 'i') ||
    ( ch == 'o') ||( ch == 'u'))  {
    try{
       throw new VowelNotAllowedException("vowel not
      allowed");
    }catch(VowelNotAllowedException e){
        System.out.println(e);
    }
    //out.write(ch);
    }
    else
      {
          outt.write(ch);
      }

  }
  outt.close();
  br.close();
  }
 catch (Exception e) {
System.err.println(e);
  }
  }
  }
```

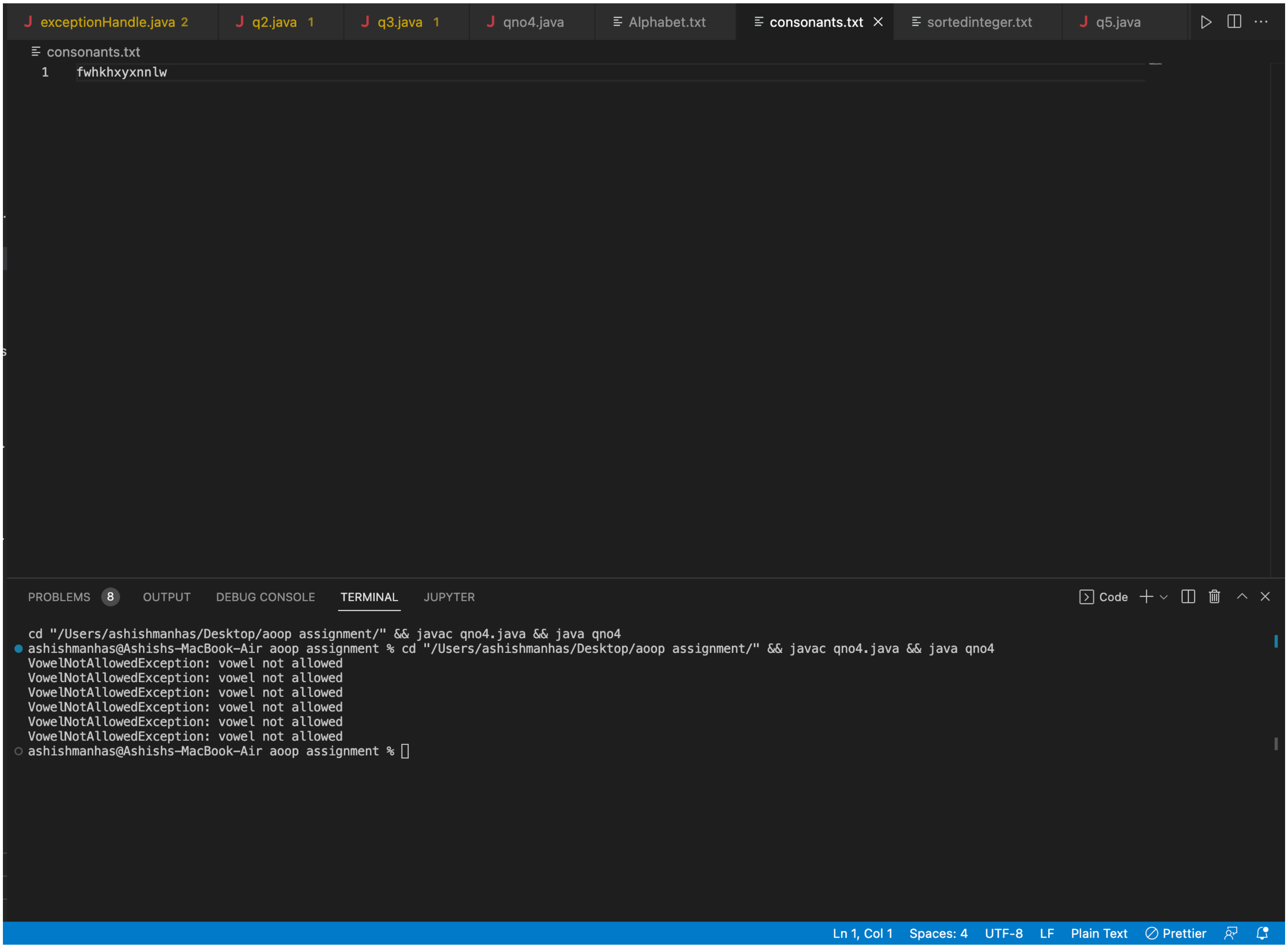
Output -:



Alphabet.txt

```
fuwhkhxuyuiuxnnilw
```

```
cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac qno4.java && java qno4
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac qno4.java && java qno4
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
ashishmanhas@Ashishs-MacBook-Air aoop assignment % 
```



```
fwhkhxyxnnlw
```

```
cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac qno4.java && java qno4
ashishmanhas@Ashishs-MacBook-Air aoop assignment % cd "/Users/ashishmanhas/Desktop/aoop assignment/" && javac qno4.java && java qno4
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
VowelNotAllowedException: vowel not allowed
ashishmanhas@Ashishs-MacBook-Air aoop assignment % 
```

5.Problem Definition:

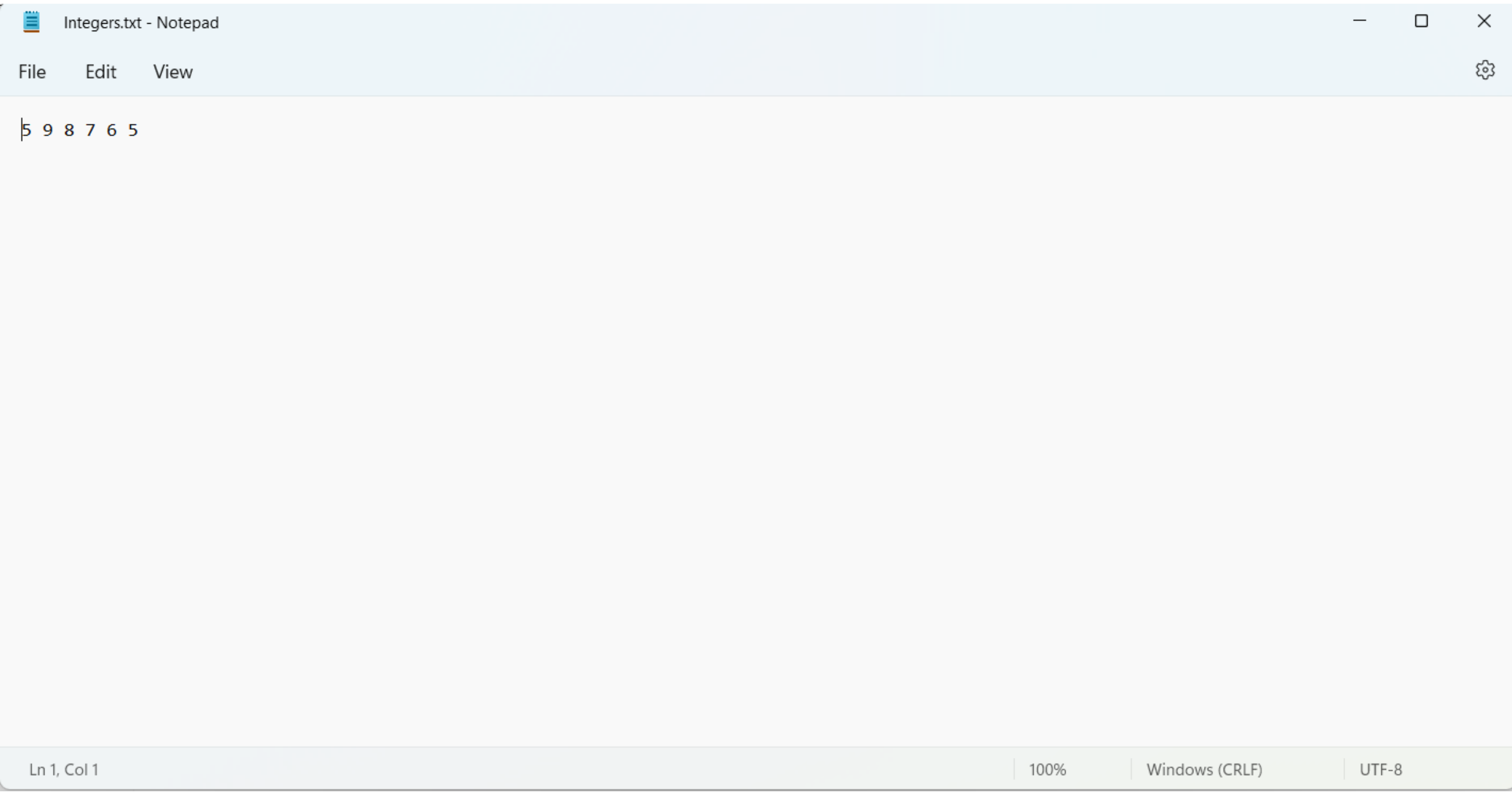
Write a Java program to implement the following scenario:

a) Create a file named Integers.txt and insert n-random integers into it
b) Create three threads T1, T2 and T3 that read n/3 integers in sequence of occurrence of numbers from the file and sort the read n/3 integers
c) Thread T4 waits for all the threads T1, T2 and T3 to complete sorting, then sorts and outputs the entire list of sorted numbers to another file named SortedIntegers.txt

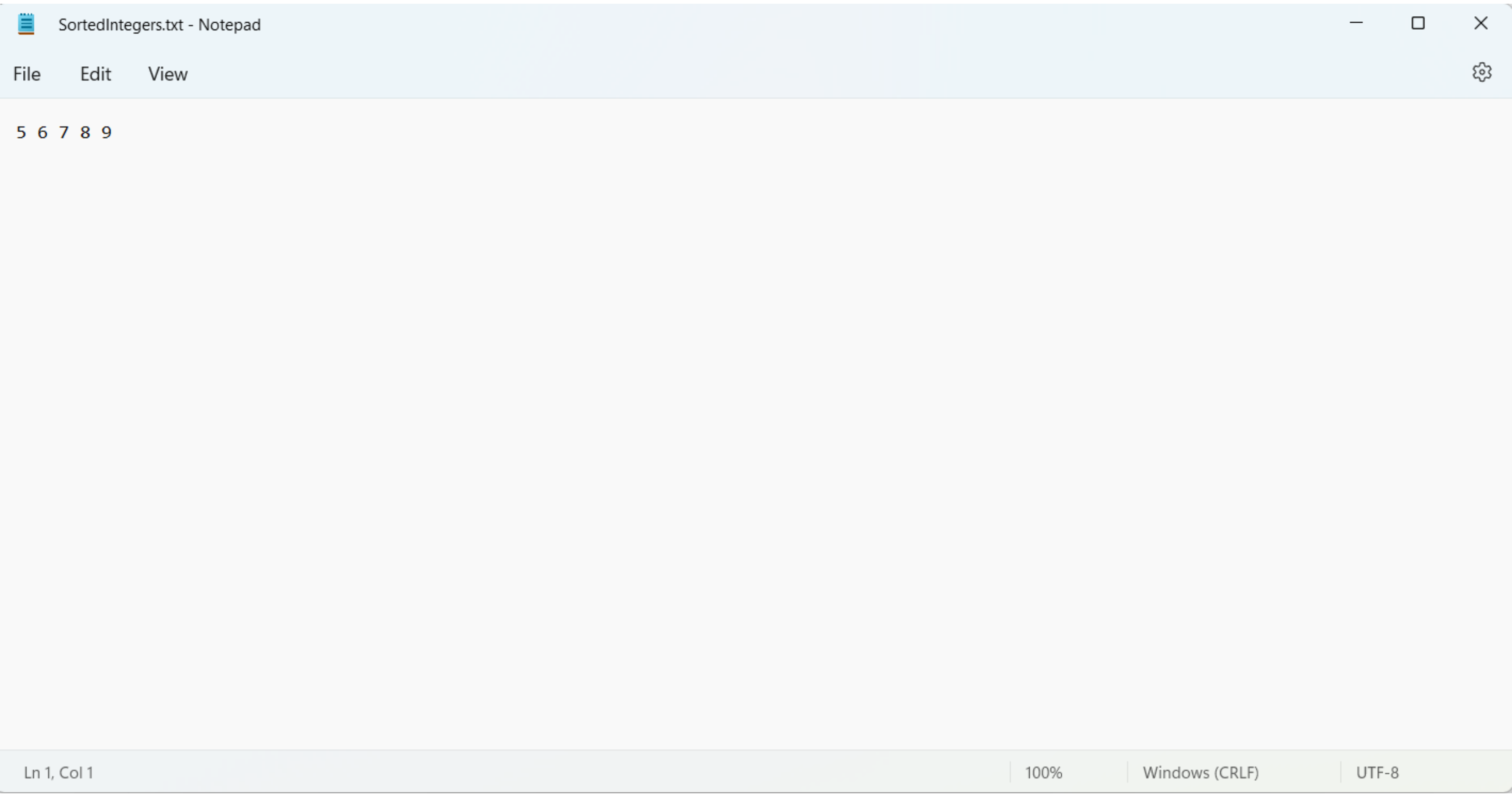
```java
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Arrays;
import java.util.Scanner;
class q5 {
private static int arr[];
public static void main(String[] args) throws
 FileNotFoundException, InterruptedException,
 IOException {
 File inputFile = new File("Integer.txt");
 File outputFile = new File("sortedIntegers.txt");
 FileWriter opWriter = new FileWriter(outputFile);
 Scanner sc = new Scanner(inputFile);
 int size = sc.nextInt();
 arr = new int[size];
int i = 0;
while (sc.hasNext())
{
  arr[i++] = sc.nextInt();

  }
  sc.close();
  Thread T1 = new Thread() {
  public void run()
 {
 ThreadSorting(arr, 0, (size / 3) - 1);
  }
  };
 Thread T2 = new Thread() {
public void run() {
ThreadSorting(arr, (size / 3), ((size / 3) * 2) - 1);
 }
 };
 Thread T3 = new Thread()
 { public void run() {
 ThreadSorting(arr, ((size / 3) * 2), (size - 1));
  }
  };
  Thread T4 = new Thread() {
  public void run() {
  ThreadSorting(arr, 0, size - 1);
  }
  };
T1.start();
T1.join();
T2.start();
T2.join();
T3.start();
T3.join();
T4.start();
T4.join();
```

```
 for (int num : arr)
 {
opWriter.append(String.valueOf(num) + " ");
 }
opWriter.close();
}
public static void ThreadSorting(int arr[], int start,
int end) {
int tempArr[] = new int[end - start + 1];
int tempIndex = 0;
for (int i = start; i <= end; i++)
 {
tempArr[tempIndex++] = arr[i];
}
Arrays.sort(tempArr); int index = start;
for (int n : tempArr) {
arr[index++] = n;
}
}
}
```

Output -:



```
5 9 8 7 6 5
```



```
5 6 7 8 9
```